# WEST Search History

DATE: Sunday, August 24, 2003

| Set Name side by side | Query | Hit Count | Set Name result set |
|---|---|---|---|
| | *DB=USPT; PLUR=YES; OP=ADJ* | | |
| L15 | L5 same (server$ near4 (load$ or workload$)) | 7 | L15 |
| L14 | L12 same (server$ near4 (load$ or workload$)) | 0 | L14 |
| L13 | L12 and (server$ near4 (load$ or workload$)) | 82 | L13 |
| L12 | l5 same (threshold$ or limit$ or max$) | 2482 | L12 |
| L11 | l9 and threshold$ | 11 | L11 |
| L10 | 5953506[pn] | 1 | L10 |
| L9 | l6 and l8 | 44 | L9 |
| L8 | (709/203 OR 709/236 OR 709/230 OR 709/231 OR 709/232).CCLS. | 3309 | L8 |
| L7 | L6[ti,ab] | 9 | L7 |
| L6 | L5 same (host$ or server$) same client$ | 154 | L6 |
| L5 | packet$ near2 (siz$ or length$ or aggregat$) | 8321 | L5 |
| L4 | L3 same client$ same server$ | 5 | L4 |
| L3 | aggregat$ near4 packet$ | 218 | L3 |
| L2 | 5197127[pn] | 1 | L2 |
| L1 | 6003089[pn] | 1 | L1 |

END OF SEARCH HISTORY

WEST

Generate Collection Print

L7: Entry 6 of 9 File: USPT Feb 29, 2000

DOCUMENT-IDENTIFIER: US 6032197 A
TITLE: Data packet header compression for unidirectional transmission

Abstract Text (1):
A broadcast transmission system transmits data packets from a server to a client over a unidirectional broadcast network. The system transmits both full-length data packets, which have uncompressed headers, and reduced-length data packets, which have compressed headers derived from associated uncompressed headers. The server compresses the data packets by compressing the packet header. Compressed packet headers contain fewer header fields than their associated uncompressed headers. The server transmits a series of intermixed full-length and reduced-length packets to the client. As the packets are received, the client determines whether the packets are full-length or reduced-length. If the packet is full-length, the client stores the uncompressed header in a header table. If the packet is reduced-length, the client rebuilds the compressed header from its corresponding uncompressed headers in the header table.

WEST

**End of Result Set**

Generate Collection Print

L4: Entry 5 of 5 File: USPT Sep 8, 1998

DOCUMENT-IDENTIFIER: US 5805823 A
TITLE: System and method for optimal multiplexed message aggregation between client applications in client-server networks

Brief Summary Text (15):
Similarly, when a single client has numerous messages to send to multiple clients, it can aggregate all of these message into as few network packets as are needed and send them to the server. The server disaggregates the messages from the packets, sending each message to the client to whom the message is destined. Again, network bandwidth is fully utilized while simultaneously optimally reducing the number of network connections necessary for communication between the clients. In accordance with another aspect of the invention then, there is provided a method for sending messages between two client applications through a central server, each client coupled to the server over a single connection.

CLAIMS:

2. The system of claim 1, wherein:

each connection between a client application and the server application has a maximum transfer unit defining a maximum amount of data for a network packet sent along the connection; and,

each client application aggregates data from messages into a network packet as messages are available for sending to the server application, and until the network packet cannot receive all of the message data of a next available message; and,

the server application creates a single network packet from the separate messages received from the first client applications that includes an amount of data equal to, or less than a multiple of the maximum transfer unit for the connection between the server application and the second client application.

# WEST Search History

DATE: Sunday, August 24, 2003

| Set Name side by side | Query | Hit Count | Set Name result set |
|---|---|---|---|
| *DB=USPT; PLUR=YES; OP=ADJ* | | | |
| L24 | 4771391[pn] | 1 | L24 |
| L23 | l20 and l8 | 12 | L23 |
| L22 | L21 and l8 | 0 | L22 |
| L21 | L20[ti,ab] | 21 | L21 |
| L20 | L19 near6 (increas$ or larger$ or big$) | 698 | L20 |
| L19 | L18 near6 (scal$ or adapt$ or adjust$ or modif$ or alter$ or chang$ or combin$) | 17855 | L19 |
| L18 | (packet$ or block$ or fram$) near6 (siz$ or resiz$ or length$ or aggregat$) | 131611 | L18 |
| L17 | packet$ near6 (siz$ or resiz$ or length$) | 10670 | L17 |
| L16 | l13 and l8 | 19 | L16 |
| L15 | L5 same (server$ near4 (load$ or workload$)) | 7 | L15 |
| L14 | L12 same (server$ near4 (load$ or workload$)) | 0 | L14 |
| L13 | L12 and (server$ near4 (load$ or workload$)) | 82 | L13 |
| L12 | l5 same (threshold$ or limit$ or max$) | 2482 | L12 |
| L11 | l9 and threshold$ | 11 | L11 |
| L10 | 5953506[pn] | 1 | L10 |
| L9 | l6 and l8 | 44 | L9 |
| L8 | (709/203 OR 709/236 OR 709/230 OR 709/231 OR 709/232).CCLS. | 3309 | L8 |
| L7 | L6[ti,ab] | 9 | L7 |
| L6 | L5 same (host$ or server$) same client$ | 154 | L6 |
| L5 | packet$ near2 (siz$ or length$ or aggregat$) | 8321 | L5 |
| L4 | L3 same client$ same server$ | 5 | L4 |
| L3 | aggregat$ near4 packet$ | 218 | L3 |
| L2 | 5197127[pn] | 1 | L2 |
| L1 | 6003089[pn] | 1 | L1 |

END OF SEARCH HISTORY

L23: Entry 9 of 12 File: USPT Dec 14, 1999

DOCUMENT-IDENTIFIER: US 6003089 A
TITLE: Method for constructing adaptive packet lengths in a congested network

Brief Summary Text (14):
A router can be used to alter packet size (i.e., reduce or increase the packet size). These altered packets can be used to bridge, for example, a 9 K Byte or 64 K Byte to an Ethernet. For example, to route between ATM and LANs, the built-in packet fragmentation capability of the router (referred to as segmentation and reassembling, or a SAR, in this example) can be used to break the LAN-sized packets into ATM cells/packets as they cross the LAN to ATM boundary. If the ATM cells are then routed back via a SAR to a LAN, each original packet is completely reassembled before being sent out over the media. Thus, even if the media is available for transmission of the data, the packet will not be transmitted until it is completely reassembled. While this method of altering the packet size allows for connection between ATM and LANs, it does not address the problem of network degradation and utilization of the available media.

Detailed Description Text (16):
Only complete packets are placed in the larger packet. In addition, the packet order is always maintained. Therefore, if the next packet eligible for combining will result in a larger packet which exceeds the maximum size allowed for packets, waste area 138 is left at the end of the larger packet, and the next eligible packet is built into the next larger packet. Waste area 138 could be large when large packets intermingle with small packets (e.g., up to approximately 1.5 k bytes). Waste area 138 is not sent in the larger packet, but it represents how much more data could be included in the larger Ethernet packet. For simplicity, received packets are not divided between larger packets. If the packets were divided, information related to upper layers (e.g., a layer three header) and regular data would have to be identified separately at a lower layer (e.g., layer two). By keeping the received packets whole, the present arrangement provides for faster processing with less errors. In the preferred embodiment, building of a larger Ethernet packet continues while the media is unavailable as long as the newly created larger packet is smaller than the network's maximum allowable packet length.

Detailed Description Text (26):
In another embodiment of the present invention, when the network experiences a medium or high load, each packet size is kept above a

L23: Entry 1 of 12 File: USPT May 13, 2003

DOCUMENT-IDENTIFIER: US 6564267 B1
TITLE: Network adapter with large frame transfer emulation

Brief Summary Text (11):
In accordance with one aspect of the present invention, a method for operating a network adapter is disclosed. According to this method, the network adapter intercepts connection negotiation packets passing between the transmission control protocol layer and a remote endpoint. The network adapter modifies the maximum segment size of the packets as necessary such that the transmission control protocol layer receives an indication that the remote endpoint has accepted a first maximum segment size for the connection, and the remote endpoint receives an indication that the host computer has accepted a second, smaller maximum segment size for the connection. Thereafter, upon receiving a request from the transmission control protocol layer to transfer an original data packet larger than the second maximum segment size to the remote endpoint over the connection, the network adapter segments the original data packet into multiple secondary data packets, each having a size no greater than the second maximum segment size. These secondary packets are transmitted by the network adapter to the remote endpoint.

Brief Summary Text (13):
Also, an article of manufacture comprising a computer-readable medium containing a driver program for a network adapter is disclosed. When executed, the driver program configures a processor to run a connection parameter manager to negotiate, for a connection, a first maximum segment size with a local host and a second, smaller maximum segment size with a remote endpoint. When executed, the driver program also configures a processor to run a context scheduler to match host-originated data packets to the connection and request a packet segmentation context on the network adapter when a matched data packet is larger than the second maximum segment size.

Current US Cross Reference Classification (4):
709/230

CLAIMS:

1. A method for operating a network adapter coupled to a host computer running a transmission control protocol layer, the method comprising: the network adapter intercepting connection negotiation packets passing between the transmission control protocol layer and a remote endpoint, wherein the connection negotiation packets can

WEST

Generate Collection Print

L16: Entry 15 of 19 File: USPT Jan 11, 2000

DOCUMENT-IDENTIFIER: US 6014707 A
TITLE: Stateless data transfer protocol with client controlled transfer unit size

Detailed Description Text (4):
The request specifies not only the file(s) to be downloaded, but also includes information that tells the server 12 how the file is to be delivered. This includes the maximum size of an individual data packet and the maximum rate at which the packets can be processed. This is to accommodate a wide range of client configurations. The server 12, upon receiving a request, may further reduce the size and rate of data packets that will be used for the transfer, depending on the current loads of both the server 12 and the network 10. The server 12 then schedules the request and begins transmitting sequential data packets, containing the client-unique transaction ID, at the appropriate rate.

Detailed Description Text (24):
blockSize--Parameter indicates a maximum PDU (i.e., data block or packet) size for the server to use.

Detailed Description Text (48):
If the request is for a file download, and the file exists, then the Request Handler 26 begins preparation for download. The server 12 must have ready access to the download data during the transfer, and therefore, files are first loaded into the cache 22 to ensure such. The file cache 22 is not required, but is consistent with the overall goal of minimizing server workload and the preferred implementation encompasses a cache. If the most recent version of the file is not already present in the cache 22, it is preloaded into the cache 22 before the transfer begins. If an older version of the file is in the cache 22 and it is not inuse, it will be replaced. If an older version of the file is in the cache 22 and is inuse, that file will be tagged for discard as soon as the current download is completed.

Current US Original Classification (1):
709/232

L16: Entry 15 of 19 File: USPT Jan 11, 2000

DOCUMENT-IDENTIFIER: US 6014707 A
TITLE: Stateless data transfer protocol with client controlled transfer unit size

WEST

Generate Collection Print

L16: Entry 12 of 19 File: USPT Jun 6, 2000

DOCUMENT-IDENTIFIER: US 6073180 A
TITLE: High-speed batch file transfer method and apparatus, and storage medium in which a program for executing the transfer is stored

Brief Summary Text (5):
Accompanying the development in the multi-media era, there exists an increasing demand for services which deliver large amounts of bulk data such as images and the like to user terminals from a server. With the current VOD (Video On Demand), due to the extreme increase in not only the transfer start orders of the file, but also control orders, e.g., pause, rewind and the like, which contribute to the load on the server processor, it is not possible to sufficiently appreciate the merits from the increased speed of the network. In contrast, for the sake of user convenience and economics, a system is being considered in which a large amount of bulk data corresponding to a single CD-ROM or movie is transmitted to a user storage medium in a few seconds to a few tens of seconds, followed by instant release of the network. The present invention provides a method and apparatus for achieving the aforementioned; the various components comprising the background of the present invention will be explained in detail below with reference to the figures.

Brief Summary Text (18):
Furthermore, with regard to the file data within contents server B101, when performing random access (i.e., not sequential access) such as rewind, fast forward or pause during the playing of image data, in addition to an excessive load which is placed on the contents server B101, the transfer speed is further reduced when performing random access from a plurality of terminals at the same time.

Brief Summary Text (20):
Further, in addition to the interface speed being limited to a low speed, when performing random access to a file such during access of an image and/or audio data file, an excessive load is placed on the contents server B101, which gives rise to further reduction of the file data transfer speed.

Brief Summary Text (30):
In addition, another object of the present invention is to allow for speed conversion of ATM links comprising different interface speeds, and, moreover, disperse large loads placed on the server by random access of large capacity files and the like, thereby

acheiving a high transfer throughput.

Detailed Description Text (69):
In general, in a server system containing a plurality of clients, when attempting to directly respond to a random access from a client, the software interruption time and load of the server processor increases tremendously such that increasing the number of clients is difficult even when the server computer is upgraded. In addition, with regard to accessing a hard disk, mechanically, the seek mode speed for assigning a position to the disk cylinder following movement of the read-write head serves as the slowest element. This seek period hence becomes the "neck" such that the throughput during random access is tremendously reduced when compared with sequential reading/writing. In the end, actual processing, i.e., data readout, is broken up and performed, such that the conversion time between processing comes to occupy a large amount of the time required for the entire process.

Detailed Description Text (77):
FIG. 14 shows the format of a UDP packet. The size of the UDP packet has a maximum variable length of 64 kB. In this case, the packet length attached to the transmission packet is mapped to the length field of the UDP header. The transmitting side top address is mapped to the unused filed of the IP header (e.g., maximum 32 bit IP option field). According to the UDP, a dummy header wherein the IP addresses of the transmission source and destination are mapped is added and a sum check of the header and all data is conducted. By means of using this sum check, it is possible to detect damage of the packet currently being transmitted. In addition, omission of the packet is determined by the non-arrival of a specific top address of the packet within a time-out period to the receiving side. Using the aforementioned, data transfer and retransmission requests are conducted according to the above-described method.

Detailed Description Text (79):
FIG. 15 shows a format of an AAL-5 CPCS-PDU. The size of the CPCS-PDU has a maximum variable length of 64 kB. In this case, the packet length attached to the transmission packet is mapped to the LI (length indicator) field of the CPCS-PDU trailer. The transmitting side top address is mapped to the unused field of the CPCS-PDU trailer (e.g., 8 bit CPCS-UU (CPCS user-to-user information)) and CPI (Common Part Identifier) field. According to the CPCS, CRC-32 (32-bit Cyclic Redundancy check) error check of a collected plurality of SAR-PDU is performed. The transfer unit is the ATM cell wherein an ATM header is attached to this SAR-PDU, and thus it is possible to simultaneously detect omission or damage of the cell by means of this CRC-32 check. Using the aforementioned, data transfer and retransmission requests are conducted according to the above-described method.

Current US Cross Reference Classification (1):
709/230

L16: Entry 12 of 19 File: USPT Jun 6, 2000

WEST

Generate Collection Print

L16: Entry 16 of 19 File: USPT Jul 27, 1999

DOCUMENT-IDENTIFIER: US 5928330 A
** See image for Certificate of Correction **
TITLE: System, device, and method for streaming a multimedia file

Brief Summary Text (11):
Many modern multimedia applications involve the transfer of a large amount of information, placing a considerable load on the resources of the network, server, and client. The use of network-based multimedia applications appears to be growing. As computers become more powerful and more people access network-based multimedia applications, there will be an increased demand for longer, more complicated, more flexible multimedia applications, thereby placing even larger loads and demands on the network, server, and client. The demand placed on servers by these ever-growing multimedia applications is particularly high, as individual servers are called upon to support larger numbers of simultaneous uses: it is not uncommon even today for an Internet server to handle thousands of simultaneous channels. Consequently, there is a need in the art for a device, system, and method that, among other things,

Detailed Description Text (16):
Pre-packetizing video information, on the other hand, may benefit from dividing, rather than merging, presentation units. For example, under the H.263 video encoding standard, video information is encoded as a sequence of video frames (i.e., a frame being a presentation unit). Although a video packet may be formed to correspond to a single video frame, or presentation unit, advantages may be attained by dividing the presentation unit into several packets. In this fashion, a large video frame may be divided into several packets so that each video packet is limited to a predetermined, maximum packet size.

Detailed Description Text (17):
By pre-packetizing the data, an appropriately designed server's processor load may be reduced by alleviating the processor from having to perform certain tasks such as constructing packets on-the-fly from the media information. With the invention, the server can simply read a packet from the file and pass it to a UDP layer of the protocol stack via a standard interface.

Current US Original Classification (1):
709/231

Current US Cross Reference Classification (1):
709/232

L16: Entry 5 of 19 File: USPT Sep 3, 2002

DOCUMENT-IDENTIFIER: US 6446132 B1
TITLE: Radio data communication system and method for carrying out data communication through a radio channel

Brief Summary Text (6):
Herein, it is to be noted that communication may not always be forcibly established on bad communication conditions, such as strong fading condition, low field intensity, in the radio communication network and a heavy load imposed on a server in the wire communication network. This is because such forcible communication under the bad conditions makes a communication time unexpectedly long and consequently gives rise to an undesirable increase of a communication charge.

Detailed Description Text (13):
Referring to FIG. 4, description will be made about a protocol OSI (Open Systems Interconnection) model of seven layers L1 to L7 and about a concept of a radio data communication method and device according to this invention. In the radio data communication device according to this invention, data communication between the radio controller 60 (FIG. 2) and the connection interface 39 (FIG. 1) is guaranteed in accordance with the radio communication protocol 672 and, likewise, data communication between the client terminal 31 and the information server 40 is guaranteed in accordance with a network protocol portion, such as TCP. Herein, it is to be noted that a heavy load is imposed on the information server as connections to the information server increase and gradually brings about a reduction of throughput processed by the use of the protocol with an increase of the load.

Detailed Description Text (19):
Furthermore, the user data length area is for arranging a user data length while the profile data length area is for arranging a profile data length, the profile data length area is divided into a packet length area for arranging a maximum size which appears on dividing the data received/transmitted and a window size representative of a packet number which can be continuously transmitted without waiting for any response from a transmitted side. The resent times area is for arranging re-send times, namely, a frequency of re-send occurrence with a predetermined time interval while the access area is for locating the number of client terminals which make access to the information server while the dial number area is for locating a connection number of the other information server to be switched while the speed area is for arranging a prediction transmission/reception data speed or rate of